**Database Systems Lab - 14CS2012**

**REGISTER NO: UR14CS228**

**DATE: 24-10-2016**

**EXPERIMENT-NO 7**

**Video Link :** <https://youtu.be/VqU3vLtdLzg>

**AIM:**

To Create Triggers based on different questions.

**DESCRIPTION:**

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.

**Program:**

1.Create a simple Trigger that does not allow delete operations on the Supplier table

CREATE OR REPLACE TRIGGER trigger1

BEFORE DELETE ON supplier

FOR EACH ROW

BEGIN

IF deleting THEN

RAISE\_APPLICATION\_ERROR(-20101,'Deletion is not enabled');

END IF;

END;

/

delete supplier Where S\_ID='ur3';

2. Create a Trigger that raises an User Defined Error Message and does not allow updating the order table.

CREATE OR REPLACE TRIGGER trigger2

BEFORE UPDATE ON order1

FOR EACH ROW

BEGIN

IF inserting OR updating THEN

RAISE\_APPLICATION\_ERROR(-20101,'Updation is not allowed');

END IF;

END;

/

update order1 SET customer\_id=10 WHERE order\_id=1000;

3. Create a trigger to implement referential integrity policy “on delete set NULL” in the order\_quantity table. That is when an order-id is deleted in the order table, set NULL for deleted order-id in the order\_quantity table.

CREATE OR REPLACE TRIGGER trigger3

BEFORE DELETE ON order1

FOR EACH ROW

BEGIN

UPDATE orderquantity SET order\_id=NULL WHERE order\_id=:old.ORDER\_ID;

END;

/

delete order1 where ORDER\_ID=1000;

4. Create a trigger to update the order table whenever the customer id is updated in the customer table.

CREATE OR REPLACE TRIGGER trigger4

BEFORE UPDATE ON customer

FOR EACH ROW

BEGIN

UPDATE order1 SET customer\_id=:new.customer\_id WHERE customer\_id=:old.customer\_id;

END;

/

update customer set CUSTOMER\_ID=40 where CUSTOMER\_NAME='john';

5. Create a trigger for customer table that inserts a newly inserted values in the customer\_audit table. The structure of the customer\_audit table is (cus\_id, cus\_name, city).

CREATE OR REPLACE TRIGGER trigger5

AFTER INSERT ON customer

FOR EACH ROW

BEGIN

INSERT INTO customer\_audit VALUES(:new.customer\_id,:new.customer\_name,:new.city);

END;

/

INSERT INTO customer VALUES (70, 'Reuben', 'Mulund', 'Mulund', 'Maharashtra','641114');

6. Create a Trigger that raises an User Defined Error Message and does not allow updating the supplier table if the supplied date is greater than the current date

CREATE OR REPLACE TRIGGER trigger6

BEFORE INSERT ON supplier

FOR EACH ROW

BEGIN

IF :new.s\_date > SYSDATE THEN

RAISE\_APPLICATION\_ERROR(-20101,'Supplied date is after today');

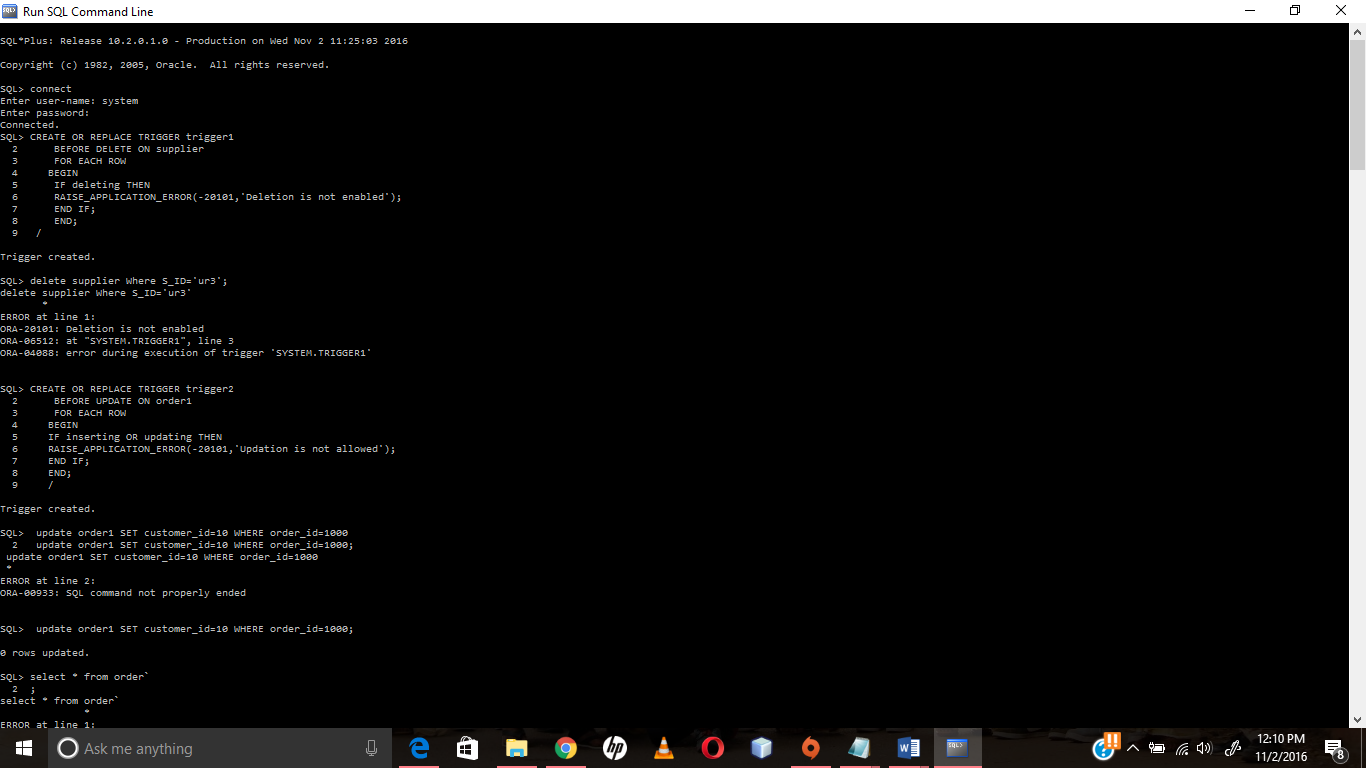
END IF;

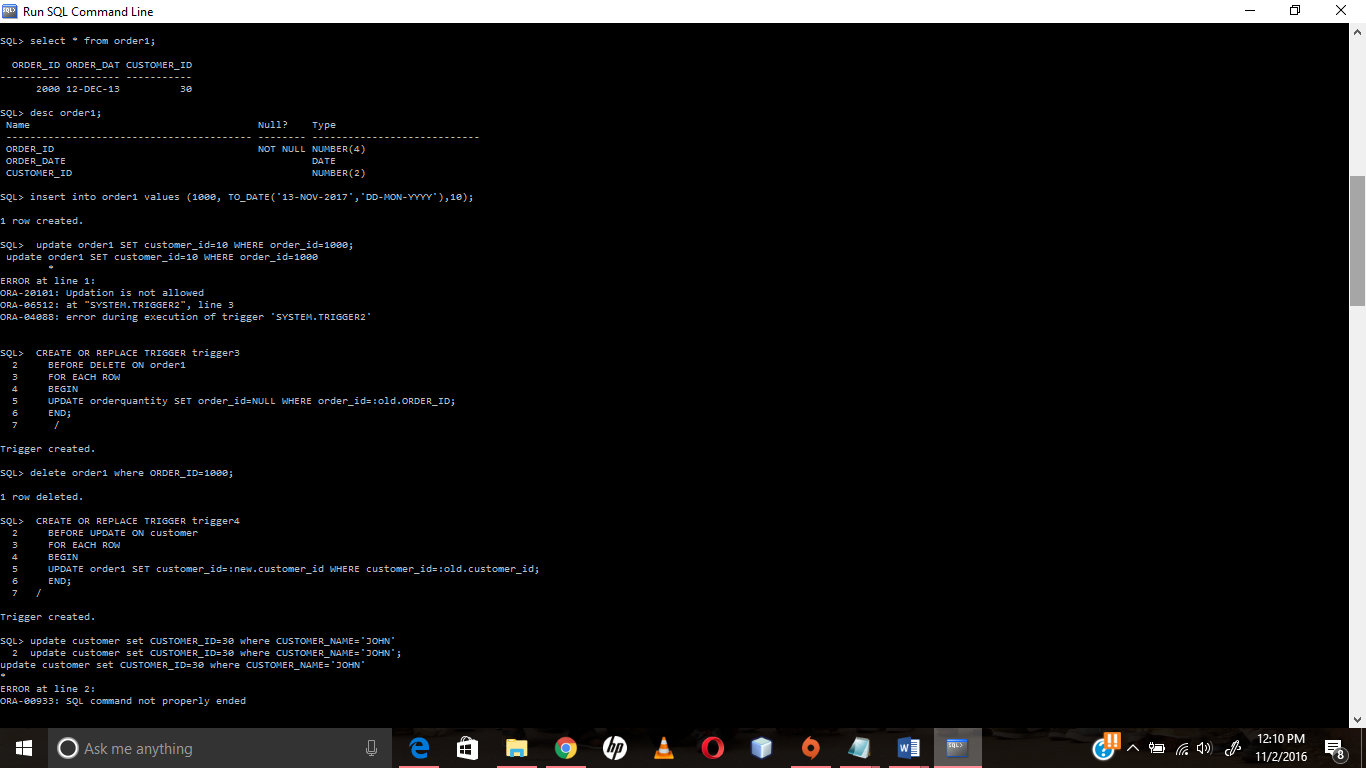
END;

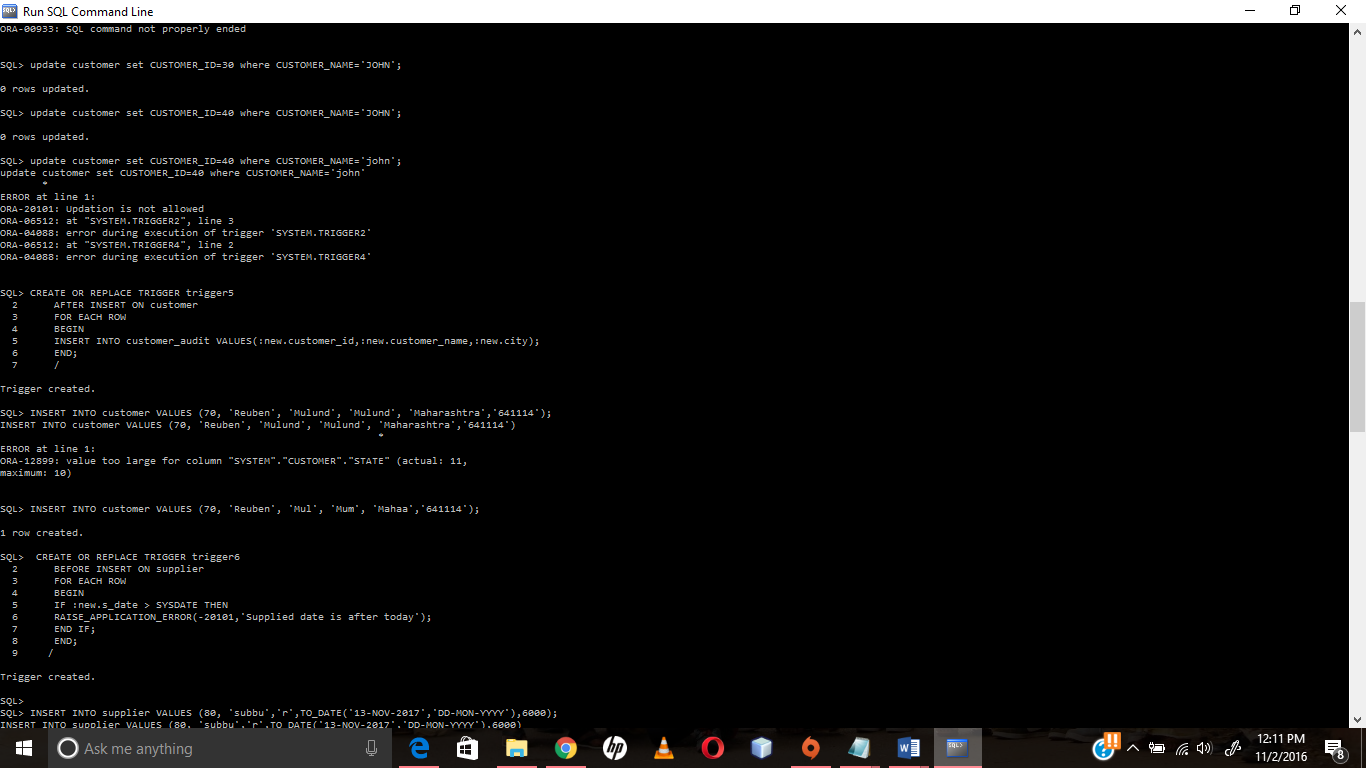
/

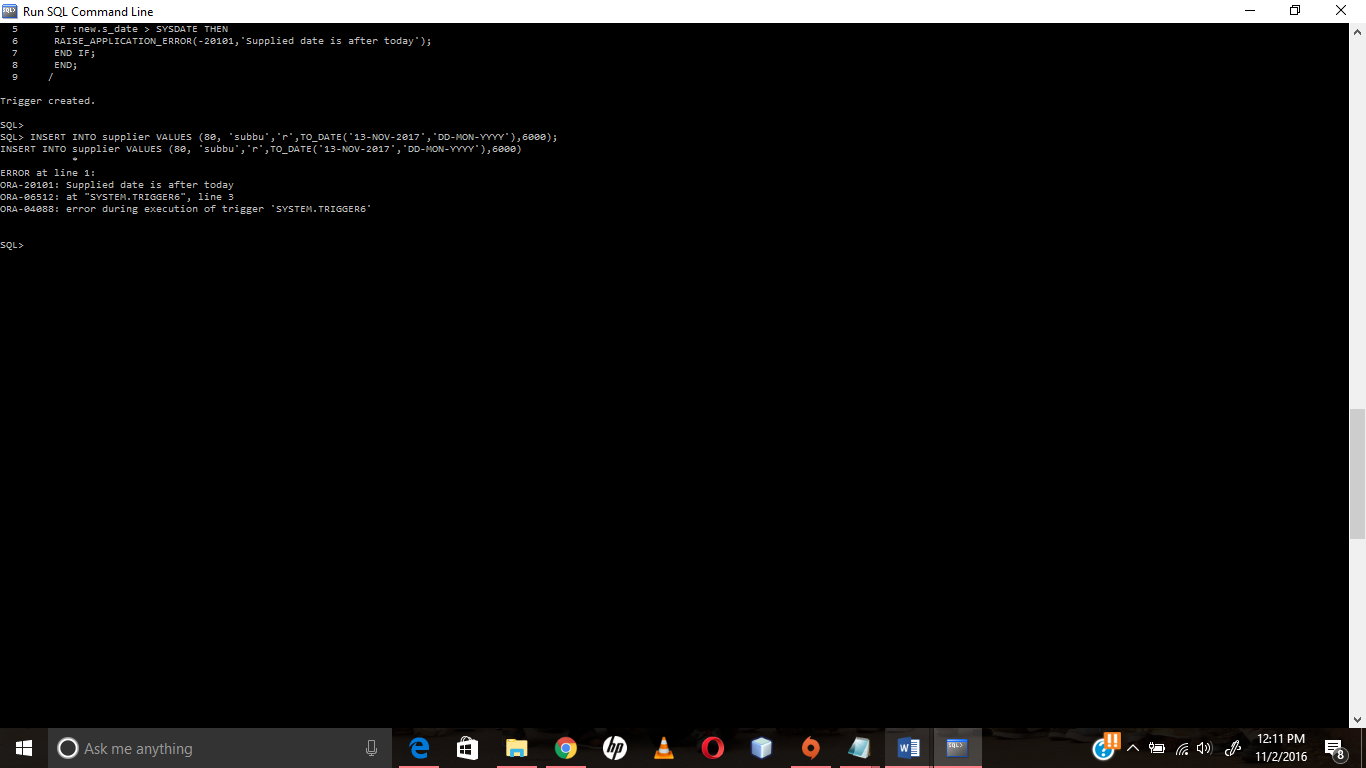
INSERT INTO supplier VALUES (80, 'subbu','r',TO\_DATE('13-NOV-2017','DD-MON-YYYY'),6000);

**Output:**









**Result:**

Triggers were successfully created and tested for all the different situations